Belhaven University
CHE321 – Physical Chemistry Laboratory
Dr. Carlson, Spring 2014

Introduction to Computational Chemistry

Introduction

Computational chemistry refers to the use of computers to perform high-level quantum chemical calculations. As you know, quantum chemistry can be quite technical and computing accurate wavefunctions for different molecules can be almost impossible to do by hand. The advanced technology we possess these days can be utilized to assist us with the calculations. Computation chemistry has reached such a level of sophistication that every chemist, chemical engineer, materials scientist, biochemist, and the like should be aware of conversant with its applications and techniques. You will be using a research grade quantum chemistry software package known as General Atomic and Molecule Electronic Structure System (GAMESS). Enough material is presented here to give a very brief overview of how this software works and why it is so powerful. It should be noted that while this is a very powerful research tool you should NOT trust it blindly. It will calculate what you tell it to without regard to how accurate or appropriate doing those electronic structure calculations may be. Each researcher must understand the underlying theory in enough detail to know the caveats associated using each method. It is responsible therefore to be critical of your results. You should also seek to validate or corroborate your results with experiments and/or other theoretical methods to further verify your data.

As you may remember from previous discussions about quantum mechanics we can express the state of atoms and/or molecules by giving their wavefunction (ψ). These wavefunctions are often found by solving the Schrodinger equation ($H\psi_n = E_n\psi_n$). It should be noted that quantum mechanics is often called quantum chemistry when applied to chemical situations or electronic structure theory because it is used to describe the electrons in these systems. You should already be familiar with atomic wavefunctions for they are where we got the ideas of s, p, d, and f orbitals. It is true that we can also find wavefunctions for entire molecules instead of just atoms. The Schrodinger equation relies on the Hamiltonian (H) operator "operating" on the wavefuntion which returns the wavefunction and the energy associated with the wavefunction. If one can accurately determine the Hamiltonian operator and the wavefunction of the system of interest then the energy can be computed accurately. The full Hamiltonian can be seen in equation 1.

$$H = -\underbrace{\frac{\hbar^2}{2}\sum_a \frac{1}{m_a}\nabla_a^2}_{\text{(1) kinetic energy of nuclei}} - \underbrace{\frac{\hbar^2}{2m_e}\sum_i \nabla_i^2}_{\text{(2) kinetic energy of electrons}} + \underbrace{\sum_a \sum_{b>a} \frac{Z_a Z_b e'^2}{r_{ab}}}_{\text{(3) potential energy of nuclei repulsion}} - \underbrace{\sum_a \sum_i \frac{Z_a e'^2}{r_{ia}}}_{\text{(4) potential energy of attraction between electrons and nuclei}} + \underbrace{\sum_j \sum_{i>j} \frac{e'^2}{r_{ij}}}_{\text{(5) potential energy of electron repulsion}} \tag{1}$$

At this point, you may be tempted to freak out a little, take a deep breath and take comfort in the fact that this is the math that GAMESS will be working with so you won't have to do it manually!

There are a few things that chemists and physicists commonly do to make working with these types of problems a little easier (or even possible in some cases). Since the Schrodinger equation cannot be solved exactly for anything larger than a hydrogen-like species (H, Li^{2+}, etc.) it is necessary to make some approximations. Now of course this can be a source of error so different approximations are made for different approaches to solving the Schrodinger equation. The most common approximations used are the Born-Oppenheimer approximation and the orbital approximation. The Born-Oppenheimer approximation (also known as the clamped nucleus approximation) involves separating the motion of the electrons and the nuclei. The nuclei are much more massive (have more mass) than electrons and so we can assume that they move much slower compared to electrons. In fact, we will assume that electrons are able to move and react so much quicker than the nuclei that the nuclei will not have time to move (rearrange) in reaction. The other common approximation is the orbital approximation which says that each electron occupies its own wavefunction called an orbital.

These approximations simplify the Hamiltonian to some extent eliminating terms (1) and (3) from equation 1. This gives the electronic Hamiltonian (H_{el}) which can be used with great success.

It should be noted there that there are other approaches one could take to find this information without necessarily having to calculate a good wavefunction for the entire system (without solving the Schrodinger equation). This is nice to learn about especially if you have a very large system with many many electrons (think on the order of a biomolecule). These cases can be approximated or strewn together from a series of parameters obtained from different experiments. This approach is known as semi-empirical. If we stick to working with the Schrodinger equation and do not use any other data (like that obtained from experiments) we call this approach *ab initio*. This is a Latin phrase meaning from the beginning. GAMESS is capable of doing both semi-empirical and *ab initio* calculations (as well as many other approaches). We might say that doing *ab initio* calculations is "expensive" compared to semi-empirical ones. It is more expensive in the sense that it requires much more computer resources (disk space, internal memory, processor time etc.).

It was mentioned earlier that some approximations are made in order to do quantum chemical calculations. These approximations introduce some error into our results. This means that no calculation can be exact. There are ways to assess which results/answers are "better" (have less error) than others. We can use the variational principle to address this issue. The variational principle shows that as long as we are using the correct Hamiltonian, the value of the total energy calculated with our "best" wavefunction will be higher than the exact energy. Hence we know that the lower the energy of a system is found to be the better the wavefunction we will have calculated. Lower energy implies better wavefunction (within the same calculation parameters). So by determining the lowest energy value we are also optimizing the wavefunction of the system. Computational chemistry software usually works by choosing mathematical functions which combine in a particular way to produce a set of molecular orbitals. These functions are then varied in a systematic way to minimize the energy thereby producing a better wavefunction for the system and a more accurate representation of how the molecules/atoms behave.

Let’s get started. First install 64 bit GAMESS (if it is not already installed) by doing the following:

1. Install "vcredist_x64" to ensure that windows visual basic c++ is up to date. You will find this file here: http://www.microsoft.com/en-us/download/details.aspx?id=14632

2. Install "mpi_x64" to ensure that windows knows about high-performance-computing and the message passing interface (MPI). You will find this file here: http://www.microsoft.com/en-us/download/details.aspx?id=14737

3. install "GAMESS.Windows.64.current" to begin the installation of gamess! the default installation directory is c:\gamess.64. You will find this file here: http://www.msg.ameslab.gov/gamess/download.html You will have to request the download and then within 24 hours (usually) you will receive a password to download the files you need.

You may also want to install some software to build molecules and input files for gamess to compute.

If so, install avogadro and/or MacMolplt both of these *seem* to do the same thing (each one has its strengths), but different people find it easier to work with one or the other. So install both!

Find MacMolPlt here: https://code.google.com/p/wxmacmolplt/wiki/Downloading you can learn more about it here: http://wiki.wxmacmolplt.googlecode.com/git/MacMolPlt_Manual.html

Find Avogadro here: http://avogadro.openmolecules.net/wiki/Main_Page

Now you have GAMESS installed on your computer, Avogadro and wxMacMoltPlt.

To start, you will use MacMoltPlt to help you build an input file (.inp) which GAMESS will calculate for you and GAMESS will compile the results in a log file (.log) which you will search for the main results of your calculation.

Ok so once you build your input file how do you run it?

To run a GAMESS input file (aka do a quantum chemical calculation!) on one of our Belhaven University department of chemistry computers – an i7 **Windows 7** PC

1. **Start button → type "cmd" and press enter** note: no need to type the quotes just what is in them
2. **type "cd c:\gamess.64" and press enter** if the computer tells you “The system cannot find the path specified” then **type “cd c:\gamess.32”** instead. Note: the 64 and 32 stand for if gamess is 64 or 32 bit.
3. **type “rungms.bat mycalculation.inp 11-64 8 0 mycalculation.log” and press enter** if you are using 32 bit gamess type **11-32 8 0** in the above command instead. If you are at a i5 computer then you should type **11-64 6 0** in the above command. How do you know? Look at the sticker on the front of the computer see if it says CORE i5 or CORE i7.
   In this command **mycalculation.inp** is the name of your input file and **mycalculation.log** is what your output file will be named. This will need to be changed to the actual name of your file that you likely made the input for on wxMacMolPlt.

These three steps will allow you to run your input file and save the output into a file called mycalculation.log in the above example. In order for this last line to work correctly you must have your input file in the c:\gamess.64 folder. This means you need to save your input file into this folder. Save it there or move it there before you run it. If you do not do this you will get an error that looks like this:

```
This job expected the input file to be in the directory:
c:\gamess.64\
Please fix your file name problem, and resubmit.
==============================================================
Now exiting.
c:\gamess.64>
```

Once you run your input file you will probably want to use another program (like Avogadro, macMolplt or wordpad (you won’t want to use notepad because it doesn’t always format it in a nice way, usually) to view your output (.log) file. Your .log file will be saved to the same folder where the input file is located (c:\gamess.64).

It is worth noting that if you run an input once a data file (.dat) will be generated in your scratch directory (that is in a folder named scr) this file saves important info about your calculation (and is often used to restart a failed calculation). This presence of this file prevents the computer from running another input with the same name unless you delete the .dat file from the scr directory. If you try to run it anyway gamess will give you an error.

Once you have finished your calculation (the command window says FORTRAN STOP) you will want to view the .log file to see what the results are. It is useful to search (Ctrl+F) for a few keywords:

Searching: **nserch** will find each step in the optimization process that you performed

Searching: **equilibrium** will find the last step in the optimization process which resulted in a minimized energy geometry

More explanation of the process follows below:

We will assume at this point that you have an input file already made and you want to have gamess compute it.

open a command prompt window: **start button → type "cmd" and press enter**
This opens a command prompt window from which you will command gamess to do you bidding [insert evil laugh].

You will desire to pass messages directly to and from the gamess folder on your computer so **type "cd c:\gamess.64" and press enter,** this will change the folder in which you are working inside the command prompt window

-you can see what is in this folder by typing "dir" and pressing enter

The rungms.bat file is used to run GAMESS. The rungms.bat requires 4 arguments.

Usage:
the command take the form "rungms.bat [input] [version] [ncpus] [ppn] [logfile]"

[input] = The filename of the input

[version] = The GAMESS version number

[ncpus] = The number of compute processes requested for this job (default: 1)

[ppn] = The number of compute processes per node (default: 0)
The default is for running on a single node (workstation). This script is not set to run across nodes. The value of [ppn] must be 0. So either pass 0 as your 4th argument or leave it blank.

[logfile] = If a 5th argument is passed then the output of the GAMESS run is redirected from the terminal window to the name of this file.

Examples:

→To run an input name sarom.inp using a GAMESS binary gamess.11-32.exe on 1 compute process with the output shown in the windows terminal, execute the following command:

```
"rungms.bat sarom.inp 11-32 1 0"
```

→To run an input name sarom.inp using a GAMESS binary gamess.11-64.exe on 1 compute process with the output saved to sarom.log, execute the following command:

```
"rungms.bat sarom.inp 11-64 1 0 sarom.log"
```

→To run an input name sarom.inp using a GAMESS binary gamess.11-32.exe on 2 compute process with the output shown in the windows terminal, execute the following command:

```
"rungms.bat sarom.inp 11-32 2 0"
```

→To run an input name sarom.inp using a GAMESS binary gamess.11-32.exe on 2 compute process with the output saved to sarom.log, execute the following command:

```
"rungms.bat sarom.inp 11-32 2 0 sarom.log"
```

→To run an input named sarom.inp using a GAMESS binary gamess.11-64.exe on 8 compute processes with the output saved to sarom.log, execute the following command:

"rungms.bat sarom.inp 11-64 8 0 sarom.log" **This is the command you will normally use**

Computational Chemistry using GAMESS – Task 1

1. Calculate the total energy of the H_2 molecule as a function of internuclear distance from values of 0.2 to 8.0 Å using the STO-3G basis set. You will use two approaches:
   a) Restricted Hartree-Fock (RHF)
   b) General Valence Bond (GVB)

   To run a GVB calculation, build an H_2 molecule in MacMolPlt. In the Control group of the Input Builder, select the SCF Run Type to be "GVB." You must also edit your .inp file. In the $SCF group, insert npair=1 $END. The new line should read (the spaces are also necessary):

   ```
   $SCF DIRSCF=.FALSE. npair=1 $END
   ```

   Save this file and then you can have GAMESS calculate it.

2. Plot the total energy as a function of internuclear distance for both the RHF and GVB methods. It will look similar to Figure 1 which is constructed for a different molecule. Notice where zero energy is on the y-axis.
3. Plot the HOMO and LUMO energies as a function of internuclear distance for both methods.
4. Construct a MO energy diagram for H_2 at its equilibrium bond distance.

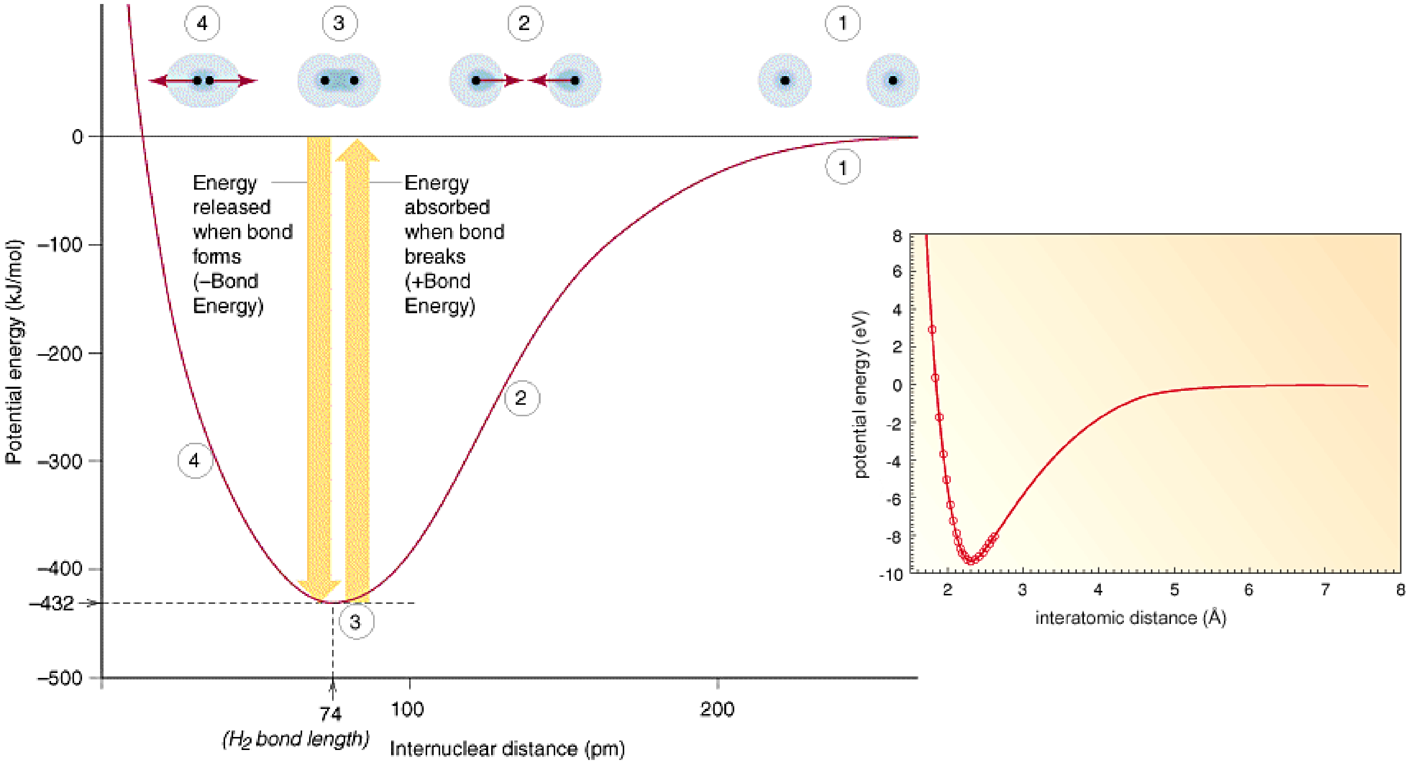
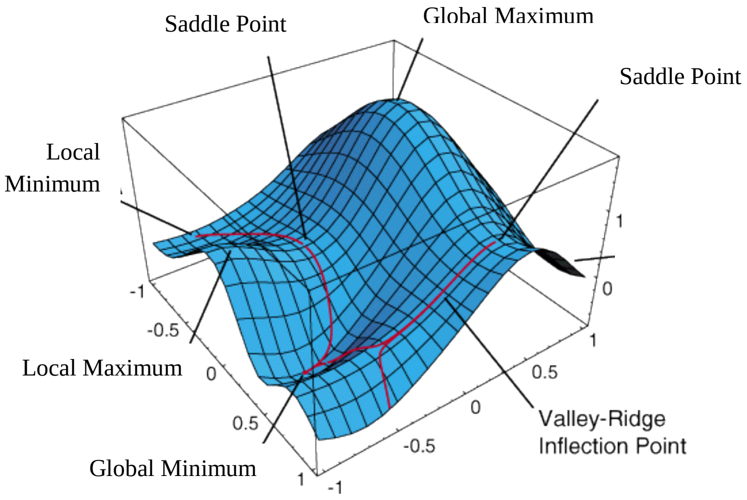
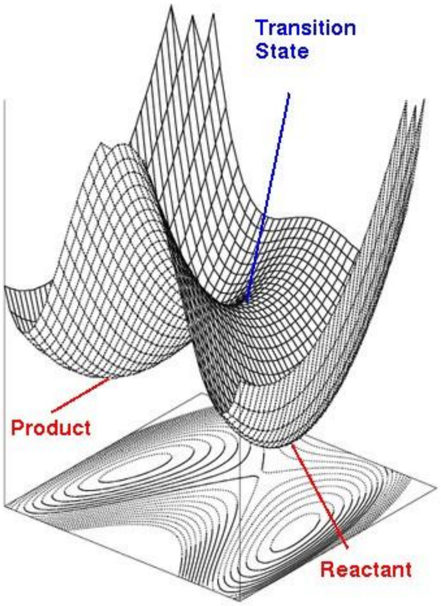


Image Source: http://www.science.uwaterloo.ca/~cchieh/cact/c120/bondel.html

Hint: I would love to see you reproduce something similar to the above graph except using your own results. You might find the "Snipping Tool" helpful as well as Powerpoint to arrange all your images into one new image.

The Potential Energy Surface (PES)





The total energy of a molecule varies as a function of the positions of all its atoms and thus is dependent on atomic coordinates. The relationship of this energy to atomic coordinates is referred to as the potential energy surface (PES). Since each position of the N atoms in a molecule can be specified in a Cartesian coordinate system, the total number of coordinates is 3N.

Each point on the PES corresponds to a specific arrangement of the N atoms in the molecule and indicates a particular molecular structure. The height of the surface at that point relates to the total energy of that structure. The example on the left shows two maxima (a local maximum and a global maximum), two saddle points, and two minima (one local and one global) on the PES. A minimum is the bottom of a valley on the PES, such that any movement away from this point gives a configuration with a higher energy. Minima occur at equilibrium structures for the system, with different minima corresponding to different conformations of a molecule. If the barrier between minima is sufficiently high, then the two structures are structural isomers of the molecule. A point that is a maximum in one direction and a minimum in all the others is called a saddle point. A saddle point corresponds to a transition structure connecting the two equilibrium structures. At both minima and saddle points, the first derivative of the energy (the gradient) with respect to every internal degree of freedom is zero. Since the gradient is the negative of the force, it means that at such points the forces are zero as well. Points at which the gradient of the energy vanishes are called stationary points. They may represent true minima or saddle points of some kind.

Geometry optimizations involve locating minima on the PES. These points represent equilibrium structures of a molecule. Optimizations can also locate transition states that may be desired or undesired. We will return to methods for finding transition states in analyzing chemical reactions. To initiate a geometry optimization with GAMESS, use the molecule builder in MacMolPlt to construct a molecule in the Master window with your best guess at its structure. In the input builder, choose the Run Type to be “Optimization” in the Control group.

The molecules we will examine are simple enough that the computational results will be minima. However, in more complex cases, this is not necessarily true. The reason for this is that geometry optimization only searches for stationary points. Thus, the computed structures may be located at a local minimum or at a saddle point. Additional steps are needed in order to determine which outcome was produced and we will not go further into this issue to determine the optimized structure.

Computational Chemistry using GAMESS – Task 2

1. Use the optimize function of GAMESS to determine the equilibrium bond distance of H_2 with the STO-3G basis set.

2. Optimize the structure of water. In order to familiarize yourself with Cartesian coordinates, you must not use the Molecule Builder in MacMolPlt. From the Subwindow menu, select Coordinates and leave the Coord. Type as "Cartesian." Add oxygen and hydrogen atoms manually and enter Cartesian coordinates for their positions. Think about the most convenient site for the oxygen atom on the coordinate system.

3. Optimize the structure of ammonia using the STO-3G basis set. You must also construct the molecule using Cartesian coordinates as you did for water. Note that if your starting guess is planar, there is a good chance that GAMESS will optimize to the planar transition state and not find the trigonal pyramidal minima as discussed above. You may wish to try this . . .

4. Optimize the structure for formaldehyde, CH_2O, with the STO-3G basis set. You may use Molecule Builder for this part.

5. Calculate the vibrational frequencies of water, ammonia, and formaldehyde using the STO-3G basis set. These computations are done on the optimized structures using the Run Type "Hessian" in the Input Builder Control group. To do this, open the .log file of your optimized structure in MacMolPlt and choose Copy Coordinates in the Edit menu. Open a new Master Windows (choose New in the File menu) and Paste (from the Edit menu, or Ctrl+V) the coordinates in the window. This should reproduce the optimized structure in the Untitled window. Open Input Builder and create a new input file to run a Hessian calculation. Note that RHF overestimates the vibrational energies. However, the general trend is correct.

6. Open the .log files of the optimized runs for water, ammonia, and formaldehyde in MacMolPlt. In the subwindow, select Frequencies to view a calculated IR spectrum of each. Select a frequency in the left hand pane of the Frequencies window. This will superimpose arrows on the atoms in the molecule shown in the Master window, designating the direction of motions. From the Master window View menu, select "Animate Mode" to see the vibration in the molecule.

Questions

1. How does the optimized bond distance in H_2 compare to the length determined last week?
2. Examine the molecular orbital of water, ammonia, and formaldehyde. Classify the MOs as π, π^*, σ, σ^*, or as a lone pair.
3. Classify the vibrations of water, ammonia, and formaldehyde as bending motions or stretching vibrations. Note that some of the motions will not fall into either of these categories. For each molecule, does the general energy trend for these modes make sense to you? The C=O stretching mode will mainly involve motion of the C and O atoms.